About the Hebrew corpora imbedded in Sketch Engine

Sketch Engine is a powerful tool for studying natural texts in context – tagged and untagged – for various purposes. Whereas originally it was designed as a tool for lexicographers, it can serve any student or researcher interested in studying language in use, whether in literature, language instruction, translation studies, or lexicology. Luckily, Sketch Engine is a multilingual environment which supports any script or writing system and provides a good support from right to left, what makes it ideal for studying Hebrew.

Here we will give a short description of the two publically available corpora in Hebrew, and present the reader with a tagger output which was used to annotate the Hebrew corpora (Meni Adler, 2007). If you need more assistance on this, you can contact noam.ordan@gmail.com.

Hebrew corpora publically available

1. *HebrewGC* – Hebrew General Corpus

This corpus was crawled from the Internet and it includes mostly newspaper materials. It is a good corpus for investigating Sketch Engine's features. Although it is a big corpus, 192,119,449 token-sized, it must be noted that it includs a non-negligible duplicate pages, which is why the results per any give query should be taken with a grain of salt. The corpus was donated by Prof Ari Rappoport and Daphna Shezaf from the Computer Science and Engineering Department at the Hebrew University in Jerusalem.

2. *HebWaC* – Hebrew WaCKy

HebWaC was crawled from the Internet. Following the WaCKy paradigm, it is a multiple domain corpus which includes blog posts, newspapers materials, commercial Internet web pages, what not. The idea behind this kind of corpus is to try to give a broad, domain-independent reflection of language. It is therefore an ideal corpus for studying "what's going on in Hebrew" in general, in a way that is not restricted to a specific text type or genre. The corpus is 60,351,738 token-sized. Duplicates were removed.

The tagging system

Most corpora in Sketch Engine are annotated for lemma and part-of-speech. It is represented vertically. Take, for example, the following sentence: "Why has no air quality test been done on this particular building since we were elected?"

This is how the three-column vertical output looks like:

```
1	Why	WRB	why
```

```
2       has     VHZ     have
3       no      DT      no
4       air     NN      air
5       quality NN      quality
6       test    NN      test
7       been    VBN     be
8       done    VVN     do
9       on      IN      on
10      this    DT      this
11      particular      JJ      particular
12      building        NN      building
13      since   IN      since
14      we      PP      we
15      were    VBD     be
16      elected         VVN     elect
17      ?       SENT    ?
```

So for each token, or surface form (the inflected word as it appears in the text), there are additional attributes: part-of-speech and lemma (the uninflected form). To take the first word as an example, the part-of-speech of 'Why' is *wh-adverb* and the lemma is 'why' (the first letter being non-capitalized). The user of Sketch Engine can conduct queries by different attributes. To take another example, if the user wished to search for the occurrences of the word 'test' as a noun (as opposed to 'test' as a verb), he could delimit the search for 'test' with the part-of-speech attribute to the tag NN (noun, singular) (line 6 in the example above).

The case is much more complicated for languages with rich morphology, like Hebrew. The tagger output in Hebrew has 28 different attributes, some of which relevant only to certain parts-of-speech.

Table 1: the tagger output for Hebrew

Attribute	Acronym	Values
Token	token	
transliteration (token)	trans	
Lemma	lemma	
transliteration (lemma)	transl	

Pos	tag	adjective adverb conjunction copula existential foreign interjection interrogative modal negation noun numberExpression numeral participle preposition pronoun properName punctuation quantifier title url verb wPrefix
pos-type	postype	amount and arithmetic-operation bracket-end bracket-start colon comma coordinating demonstrative determiner dot exclamation-mark gematria hyphen impersonal literal-number numeral-cardinal numeral-fractional numeral-ordinal or other partitive personal proadverb prodet pronoun question-mark quote reflexive relativizing semicolon slash subordinating yesno
prefix string	prestring	ב בכ ו וב ובכ וכ וכש וכשל ול ומ ומכ ומש וש ושב ושל ושמ כ ככ כש כשב כשל כשמ ל לכ לכש מ מכ מש משב משכ משל משמ ש שב שכ שכש שכשמ של שמ שמש
base string	basestring	
suffix string	sufstring	גם ה הם הן ו י ך כם כן ם ן נו
Gender	gender	feminine masculine masculine-and-feminine
Number	number	dual dual-and-plural plural singular singular-and-plural
Status	status	absolute construct
Polarity	polarity	negative positive
Person	person	1 2 3 any
Tense	tense	beinoni future imperative infinitive past
Binyan	binyan	Hifil Hitpael Hufal Nifal Paal Piel Pual
prefix conjunction	prefconj	conjunction
prefix definite article	prefdefinite	definiteArticle
prefix interrogative	prefinterrog	
prefix preposition	prefprep	preposition
prefix subordination conjunction / relativizer	relativizer	relativizer/ subordinatingConjunction
prefix temporal subordinating conjunction	preftemp	temporalSubConj
prefix adverb	prefadv	adverb
suffix function	suffunction	accusative-or-nominative possessive pronomial
suffix number	sufnum	feminine masculine masculine-and-feminine
suffix gender	sufgender	plural singular
suffix person	sufper	1 2 3

Whenever an attribute is irrelevant or missing for a token it was replaced with the string 'NULL'. Thus, if you want to search for all nouns in the corpus which are prefixed with the definite article, you conduct the following CQL query:
[tag = "noun" & prefdefinite = "NULL"]. Read more about CQL here.

Note that *tokens*, *lemmas*, *prefixes* and *suffixes* were transliterated according to the following key:

א	ב	ג	ד	ה	ו	ז	ח	ט	י	כ	ל	מ	נ	ס	ע	פ	צ	ק	ר	ש	ת
a	b	g	d	h	w	z	x	v	i	k	l	m	n	S	y	p	c	q	r	e	T

Whereas the *token* and *lemma* attributes are available both in Hebrew alphabet and Romanized transliteration, the *prefix string* and the *suffix string* appear only in transliteration.